

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.**

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
27 September 2001 (27.09.2001)

PCT

(10) International Publication Number
WO 01/71551 A1

(51) International Patent Classification⁷: G06F 17/24, 17/30

(74) Agent: THRASHER, Steven, W.; Jackson Walker L.L.P., 2435 N. Central Expressway, #600, Richardson, TX 75080 (US).

(21) International Application Number: PCT/US01/09308

(22) International Filing Date: 22 March 2001 (22.03.2001)

(25) Filing Language: English

(26) Publication Language: English

(81) Designated States (*national*): AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZA, ZW.

(30) Priority Data:
09/532,713 22 March 2000 (22.03.2000) US

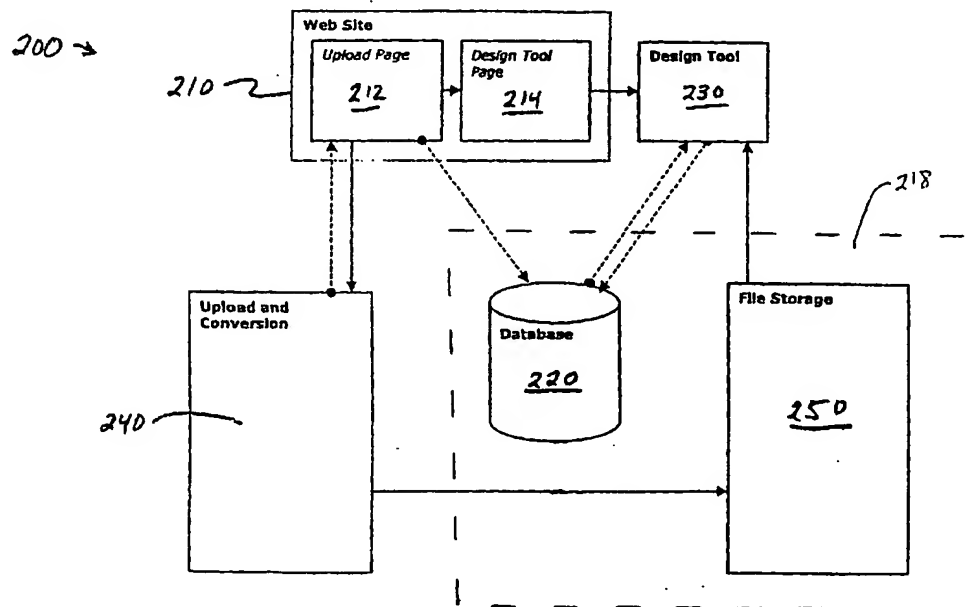
(84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

(71) Applicant: HCA MARKETING INC. [US/US]; 5931 Melshire Drive, Dallas, TX 75230 (US).

(72) Inventor: COLEMAN, Jason; 920 Mountain Terrace, Hurst, TX 76053 (US).
Published:
— with international search report

[Continued on next page]

(54) Title: SYSTEM AND METHOD FOR CHANGING ITEMS ON AN INTERNET WEB SITE



(57) Abstract: The present invention provides systems and methods of changing an item on a web page. The method generally loads a design tool from the web page onto a client-machine, requests a change to an object (either uploaded to, or selected from, the web page), selectively downloads elements necessary for accomplishing the requested change to the item in the client-machine, and then performs the actual change to create an altered item. A system of altering an item on a web page includes a web site, an upload system coupled to the web site, a design tool in communication with the web site, and a memory in communication with the design tool, the web site, and the upload system.

WO 01/71551 A1



For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

SYSTEM AND METHOD FOR CHANGING ITEMS ON AN INTERNET WEB SITE

5 FIELD OF THE INVENTION

The present invention relates generally to computer-based commerce tools, and, more particularly, to systems and methods for changing the attributes of text or graphics associated with merchandise purchased over the Internet.

BACKGROUND OF THE INVENTION

As innovations in the Internet have enhanced the visual experience for users of the Internet, the Internet has facilitated the sale of customized items, such as merchandise, graphics, text, or art work, for example. The sale of customized items requires vendors to supply customers with accurate, high quality images of the merchandise they may order and the customizations (or changes) the customers may apply.

A design tool is a software application that is used to modify the appearance of an item on a web page. In general, a design tool provides a user the ability to view an item from multiple points of view, and alter various attributes of the item, such as the items color, material, dimensions, for example. Ultimately, a design tool should provide a user the visual feedback needed so that the user may determine whether or not to make a purchase. Accordingly, the design tool should present merchandise with a fidelity that is sufficient for the user to achieve meaningful understanding of what the final physical product will be like. Furthermore, it is desirable that a design tool provide a user with nearly instant visual feedback of the

modifications that they are making to the merchandise, so that a user will become a customer.

5 To provide high quality images, vendors of customized items need design tools that provide visualization and customization abilities in an on-line environment. The basic method for customizing merchandise includes choosing attributes for the item, including the item's color, material, dimensions, for example, as well as the color, material, and dimensions of the item's components. Given that customized products frequently include
10 the application of customized text or customized art work, any Internet web page design tool should support the application of both text and art work. Given the need for fast response time, the dynamic application of product customization across the Internet using the prior art is particularly problematic.

15 One issue associated with product customization over the Internet is the need to reload a web page each time a single change is made to an item. Accordingly, figure 1 (Prior Art) is a flow chart of a modification algorithm 10 associated with a prior art method of changing the appearance
20 of an item on a web page. First, in a load web page step 20, a web page, typically located on a remote server, is down-loaded into a client-machine (meaning the personal computer or other computing device the user is using to alter the item). Generally, the web page will have an item which is to be changed and form element associated with it.

25 Form elements are typically icon, menu, or text-button driven commands which can change the appearance of the item. For example, one form element may allow a user to select the texture of the item (or a portion of the item). Likewise, another form element may allow the user the ability
30 to select a color for the item (or a portion of the item). Accordingly,

following the load web page step 20, the user will change the item some way in a change item step 30. In the change item step 30, the user may change the appearance of the item or alter some customizable attribute of the item, such as the items texture or material.

5

After making a change to the item, the modification algorithm 10 proceeds to a server processing step 40. In the server processing step 40, the web server uses a server - side software application to implement the desired change in order to build a new visual representation of the altered item. Following the server processing step 40, the server-side software application builds a new web page in a build new web page step 50. The new web page is built in order to display the new visual representation of the altered item. Then, in a load new web page step 60, the new web page is loaded onto the client-machine and the changed item is deployed.

15

If additional alterations are desired to change the appearance of the item, those additional alterations of the item can be achieved by following the above steps. Accordingly, in an additional customization step 70, the user may choose whether or not to make additional alterations to the item. If the user desires to make additional alterations to the item, the modification algorithm 10 returns to the change item step 30. If, however, the user does not desire to make additional alterations to the item, the modification algorithm 10 proceeds to an end step 80, and no further changes or alterations are made to the item.

25

Accordingly, each time a user wishes to make a single change to an item, the modification algorithm 10 is repeated. This means that each time the user wishes to add a new font, or change the color of even one pixel, a new image and (Hypertext Transfer Language) HTML page must be generated. This results in a new page download requirement for each

30

incremental incorporation, change, alteration or customization of the item. Thus, the user must download a number of web pages equal to the number of changes they wish to make to the item. Accordingly, the speed of the process is directly related to the number of customizations and changes in
5 operations that user wishes to make. This is inefficient. Therefore, what is needed is a system and method of changing an item that allows for continuous, real-time feedback and realistic item representation to the user.

SUMMARY OF THE INVENTION

The present invention achieves technical advantages as a system and method of altering an item displayed on a web page. The system is
5 comprised of a software module called a Design Tool. The Design Tool is embedded in a web page which, by downloading once into a client-machine, supports the ability to make multiple changes to an item without re-loading the web page. When initially loaded into a client-machine, the Design Tool contains an image or images of an item (such as a product), as
10 well as controls for making changes to the item or items. The Design Tool provides the ability to download elements on an as needed basis in the background (preferably without the user's knowledge of the download). In addition, the system provides an upload system capable of transferring an item to be changed from the client-machine to the web page, as well as
15 memory for storing items to be altered, as well as information regarding the items. In general, the method loads the Design Tool from the web page onto a client-machine, receives a request for changing the item, loads the elements needed to alter the item, and changes the item according to the request for change. Because the changes to the item are accomplished on
20 the client-machine, visual feedback of the customized item is refreshed in real-time so that most every change is visible instantaneously to the user.

Although the new advantages of the present invention are too numerous to discuss fully here, several advantages are of particular interest.
25 First, the Design Tool executes on a client-machine, more form elements and controls may be provided to the user due to the fact that additional Design Tool functionality will not affect the speed at which a web page may be reloaded (or refreshed). Second, client-machine base applications are less likely to experience performance deterioration due to Internet
30 traffic or connection bottlenecks. Third, the present invention provides real

time, photo-realistic, images which may also be altered, rotated, or otherwise changed in real time. In addition, the overall download performance of the present invention is superior to prior art solutions by a factor equal to the number of customizations made by a user, since only a single web page must be downloaded for any number of customization steps with the present invention, while the prior art requires an additional web page to be downloaded for each alteration. Other advantages of the present invention should be apparent to those of ordinary skill in the art.

In one embodiment, the invention is a method of changing an item displayed on a web page. The method generally includes the steps of loading a design tool from the web page onto a client-machine, requesting the change of the appearance of an item displayed in the design tool, selectively downloading elements necessary for accomplishing the requested change to the item (from the web page to the client-machine), and then implementing the change to the item in real time on the client-machine to create an altered item.

In another aspect the invention includes a method of transferring to web page an item to be altered. This embodiment includes the steps of uploading an item from a client-machine to a server, converting, if necessary, the format of the item to a format compatible with a design tool, and saving the item in a file storage area.

In yet another embodiment, the present invention is a system of changing an item on a web page. The system includes a web site, an upload system coupled to the web site, a design tool in communication with the web site, and a memory in communication with the design tool, the web site, and the upload system.

BRIEF DESCRIPTION OF THE DRAWINGS

Various aspects of the invention, including alternative embodiments, can be identified by reference to the following Detailed Description of a Preferred Embodiment. Likewise, the Detailed Description of a Preferred Embodiment can be better understood by
5 reference to the drawings, in which:

Figure 1 (Prior Art) is a flow chart of a modification algorithm associated with a prior art method of changing the appearance of an item on
10 a web page;

Figure 2 illustrates one embodiment of an architecture for the design system of the present invention;

Figure 3 is a flow chart of an upload algorithm which may be used
15 to upload an item to the Design Tool web page; and

Figure 4 is a flow chart of a customization algorithm used to change
20 an item in the Design Tool.

DETAILED DESCRIPTION OF A PREFERRED EMBODIMENT

The present invention provides users with real-time visual feedback of changes made to items, such as merchandise or any customizable product or service that can be visually represented or viewed over the Internet. The system provides a flexible architecture configured to enable real-time visual feedback of changes made to an item over the Internet via a web page-embedded designed tool that is downloaded to a client-machine. Likewise, the method, in one embodiment, outlines the process flow for providing real-time visual feedback of item changes.

A System Architecture

A number of components interact to provide functionality to a Design Tool, and are known collectively as a design system. In general, a design system employed by the present invention will comprise a web site, a Design Tool, an upload module, and a memory having a data base and a file storage area. Figure 2 illustrates one embodiment of an architecture for the design system 200 of the present invention.

A web site 210 functions as the basis for a user, (who could be a potential customer), interaction with the design system 200. The web site 210 includes web pages, typically written in HTML, that host a Design Tool 230, and the web site 210 has at least one page that is integrated with an upload module 240. Web site 210, includes at least a Design Tool page 214, and an upload module page 212. Preferably, the upload module 240 is built in ASP and communicates with the data base 220 through the web site 210. The user will see that the Design Tool 230 has all of those features associated with prior art design tools. Other media that can execute applications over the Internet include alternate platforms such as PDAs, Windows CE based platforms, and cellular telephones, for example.

The Design Tool 230 is a client-side application that is downloaded to the user web browser through the Design Tool page 214. The Design Tool is embedded in a web page, and provides the features of real-time visual feedback for changes made to items, the ability to upload items from a client-machine, the inclusion of standard items (such as art work and fonts), and allows a user to alter the appearance of the items color, alignment, font, text, size or other attribute. In addition, the Design Tool should allow for real-time dynamic sizing, positioning, and rotation of an item. Furthermore, specific construction materials may be designated for the object, and specific attributes of the material, such as its color or texture, for example, can be indicated. The present invention provides all of these functionalities and allows a user to make virtually unlimited changes to an item, while requiring the user to download only one web page.

The Design Tool 230 is preferably a Shock Wave Application (SWF or "Flash"). Flash is an embedded multimedia format that can be played directly in a web page browser. Flash is preferred because it delivers high fidelity content at a low-bandwidth cost and can deliver an application in a small file size. Furthermore, ActionScript, the native scripting language of Flash, provides a language in which a design tool may be written. Accordingly, ActionScript enables the Design Tool 230 to be configured to dynamically manipulate text, objects, and other attributes of items. Of course, other types of script may be used and other applications may be employed to build Design Tool 230.

Another advantage provided by building the Design Tool 230 with Flash is that Flash can be authored modularly so that multiple SWF files can be embedded in other SWF files, thus allowing the embedding of additional fonts and objects into the Design Tool. In addition, Flash may

be streamed, meaning that a SWF file may be displayed in a web page browser even though parts of the SWF file are not yet completely downloaded. This allows additional design tool file components to be downloaded without a user being aware that the download is taking place.

5 In addition, streaming allows the Design Tool 230 to add additional files to the client-machine without loading additional web pages.

The Design Tool 230 can be created in any self-contained, low bandwidth media that is capable of interactivity and scripting features (similar to Flash). Two such platforms capable of providing similar features are JAVA and Visual Basic (VB). However, applications created in JAVA or VB or similar platforms that perform the same or similar tasks as the Design Tool typically have large file sizes. Thus, they are typically prohibitively large for acceptable download performance. However, as

10 Internet access bandwidth increases over time, it is conceivable that in the future Internet connection speeds may be capable of supporting design tool applications developed in JAVA, VB, or other similar platforms available for the design tool process. In addition, it is conceivable that software manufacturers and developers will distribute new applications (similar to

15 Flash) capable of implementing the Design Tool 230.

20

The upload module 240, implemented specifically in the present invention as an Upload and Conversion module is typically a server-side software program. The upload module 240 comprises applications that

25 allow a user to have the capacity to transfer items, such as image files, stored in the local client-machine memory, to memory 218 at the server (such as file storage 250). The upload module 240 may convert an uploaded item to a format compatible with the Design Tool 230 so that the items may be changed in the Design Tool according to user preferences.

30

More specifically, the upload module 240 is part of the vendor web site, and allows a user to browse their local computer client-machine to select an item for uploading to the web page. The web page 210 then stores the item in a section of memory 218 known as file storage 250 and
5 associates the item with the user who uploaded it in a database 220. Preferably, the item is stored in a format which can be embedded in the Design Tool 230.

Accordingly, the upload module 240 includes a conversion
10 application to convert files from known formats, such as JPG, TIF, for example into a format which is embeddable into the Design Tool 230, such as Flash. The Upload and Conversion module uses Macromedia Generator to embed graphic images into a flash format, while the uploading and conversion of an item is achieved in three distinct parts: an upload and
15 storage stage, a conversion to PNG format stage, and an embedding into SWF formats.

In the upload and storage stage, Web Super Goo's Image Goo is employed as the server- side component of the upload and storage process.
20 The upload and storage process is driven by the web page's ASP (Active Server Page) script. When Image Goo completes the upload and storage process, it then starts the conversion process. The conversion process generates a Portable Network Graphic (PNG) version of the file. The PNG format is needed because it is capable of supporting a transparency attribute
25 when it is imported into Flash.

Accordingly, Image Goo opens a file to be converted, sets a background color to be the "transparent" color, and saves the file in a PNG format. When the upload and storage, and conversion processes are
30 complete, the ASP page is notified by Web Super Goo. Then, the ASP

page will call Macromedia Generator to begin the embedding process. ASP code in the web page instructs Macromedia Generator to use specific templates and to embed specific files.

5 Macromedia Generator then creates a new SWF file and embeds the uploaded item into the SWF file. The Macromedia Generator saves the SWF file in memory 218 in file storage 250 and notifies the ASP page that the item has been stored in file storage 250. The ASP page then records the path name and ownership information regarding the uploading item in the
10 data base 220 contained in memory 218. The ASP page also notifies the user that the process of uploading and storage is complete and may allow the user the opportunity to upload another item or to navigate to other parts of the web page. Accordingly, the SWF files created by Macromedia
15 Generator are ready to be inserted into the Design Tool 230.

15 Of course, the Macromedia Generator may in future versions have its functionality extended to achieve all of the above processes. It is also possible that other software manufacturer may distribute applications that are capable of performing some or all of these processes as well. In
20 addition, Web Super Goo and Macromedia Generator are capable of being controlled by means other than an ASP page. For example, they could be driven by PERL or other CGI language.

25 Memory 218 provides the database 220 and the file storage 250. The database 220 is capable of storing information, and is used to store information about the uploaded items, customer accounts, customized items, such as merchandise, created by the Design Tool 230. The file storage 250 is used to store files that may be needed by the Design Tool 230. In addition, the file storage 250 holds the font files and image files as

well as the art uploaded by a user so that these items may be pasted to the upload module 240.

5 It is desirable that as items are stored, that they can be changed into a standardized file format which can be used by the vendors who actually create merchandise. Of course, this data could also be transferred to a proprietary format if the vendor so desires. Accordingly, items should be capable of being stored in formats traditional in the digital printing industries, bordering industries, screening, engraving, embossing and
10 debossing, dye cutting, and other manufacturing industries. Furthermore, it is desirable that files, such as font files, be capable of downloading to a client-machine on an as needed basis and of downloading in the background while the Design Tool 230 is running on the client-machine. Accordingly, font files and image files should be stored as modules.

15 In addition, art files may have colors which are associated, or indexed, to be transparent—this means that they display on top of the product image without displaying the rectangular background shapes that accompany any type of raster (BMP type) image. Also, specific type styles
20 of a particular font will have their own module meaning that a font's italic, bold, underline, and bold-italic styles will each have their own module.

The LoadMovie function of Shockwave Flash allows a SWF file to be dynamically loaded into other SWF files. Accordingly, user may control
25 the type and specific source of image files they would like to embed through applications developed in Action Script.

The file storage 250 stores media files in the original uploaded format as well as other formats used throughout the conversion process.
30 The file storage area 250 does not have to be a single device or depository

for files, and, in fact, may function more efficiently if it is made up of several unique depositories. For example, a separate file server may be used to implement a portion of file storage 250 used to store uploaded image items. Then, as the size of memory used by certain types of items grows, additional servers may be added to implement the file storage 250.

Database 220 is preferably implemented as a Microsoft SQL server database. To convert items into a format useable by a vendor, several applications must be employed. Accordingly, Macromedia Generator is used to dynamically build graphic files based on input data (information about the item). However, Macromedia Generator creates content for the Internet, it is not enabled to create the formats desired by vendors, such as encapsulated postscript EPS, PDF (Portable Document Format), AI (Adobe Illustrator), or PJTF (Portable Job Ticket Format), for example.

An offline system including Macromedia Generator and other software is used to convert the data recorded by a customer into a format usable by a printer or manufacturer using templates appropriate for the item type. Accordingly, the SWF file that is produced by the Macromedia Generator is saved in file storage 250. Next, an automated version of the Flash Editor is employed to convert SWF format files into EPS. The Flash Editor creates a new FLA file, and imports the SWF file then exports the EPS file. The EPS file is specified in this example because it is a widely accepted format used by graphic professionals, and is readily adaptable to printers. However, it should be understood, that a SWF formatted file may be transferrable into other formats.

A Method of Implementation

The present invention allows the user to customize an item by changing (altering the appearance of, or adding art or textual objects to) an

item selected from a catalog on the Design Tool web page, or which is uploaded from a client-machine. Figure 3 is a flow chart of an upload algorithm 300 which may be used to upload an item to the Design Tool web page running on a server. In general, the upload algorithm 300 uploads an item as a computer file, converts the file into a format embeddable into the Design Tool, and then saves the file into the design system memory.

More specifically, the upload algorithm 300 begins with a locate file step 310 in which the user locates on the client-machine a file the user would like to use as an art object or item in the Design Tool. Next, in an upload file step 320, the file is specifically transferred from the client-machine to the server running the design system, and the file is sent to the upload module. The uploaded file step 320 may load a file directly from the client-machine to an upload module 240, such as the Upload and Conversion module.

Next, the upload module converts the format of the uploaded file in a convert format step 330. In the convert format step 330, the uploaded file is altered from its original format into a format which is embeddable by the Design Tool 230. Once the uploaded file is converted into a format usable by the Design Tools, the file compatible with the Design Tool is stored in file storage in memory in a save file step 340. Next, in a notification step 350 the upload module is notified that the transfer of the item is complete. Then, following the notification step 350, data about the item is saved in a database in a save data step 360. After data about the item is saved in a database, the user is informed that the transfer of the item from the client-machine to the web page is complete in an inform user step 370.

A method of changing an item through a Design Tool located and running on a client-machine may be better understood by expressing the

method as an algorithm. Figure 4 is a flow chart of a customization algorithm 400 used to change an item in the Design Tool. First, in a select item step 410, a user will either choose to change an item he has uploaded or may choose to change an item which already exists on the web page.

5 Once the item to be changed is selected in select item step 410, the customization algorithm 400 proceeds to a load web page step 420. In the load web page step 420, the user accesses the Design Tool page 214 which automatically loads the Design Tool 230 onto the client-machine the user is using then a load Design Tool 430. Next, in a change item step 440, the

10 user may use the Design Tool controls to make changes to the item.

Next, in a load element step 450 the Design Tool cooperates with the web page to selectively download elements (such as modules) necessary for accomplishing the requested change to the item. Accordingly, these

15 elements are transferred from the web page to the client-machine. Then, in a client processing step 460, the Design Tool 230 running on the client-machine implements the changes to the item requested by the user to create an altered item. Thus, the changes are immediately made to the item, and the user of the Design Tool immediately sees the changes on the client-

20 machine without a new web page downloading.

If the user desires to make an additional change to the item, all the user has to do is make the change and the customization algorithm 400 will automatically process the change request by automatically returning the

25 change item step 440, and the additional changes will be processed, and indicated by the no, or "N" decision in a satisfied step 470. If the user is satisfied with the item at this point, thus desiring no further changes, the user may indicate so by entering in an indication of satisfaction by actuating an icon, selecting a menu command, or some other means. Alternatively,

30 user satisfaction with the item may be assumed if the user does not interact

with the item for a predetermined period of time, such as three minutes. This decision by the user to not make additional changes to the item is represented as a "Y" in the satisfied step 470. Following the satisfied step 470, if the user is satisfied with the item, the customization algorithm 400 proceeds to a load item to server step 480.

5

In the load item to server step 480 the altered item with all of its changes is loaded into the web page in a single step. Then, the web page may be refreshed in the browser on the client-machine. Accordingly, the present invention displays and makes all user changes to an item, and displays those changes to the user while requiring the loading of only one web page (whereas, in the prior art, a new web page would have to be loaded following each incremental change made to the item).

10

To quickly and efficiently transfer fonts between the web page and a client-machine, the modular features of Flash are adapted to work with the font face control of the variable field to emulate RTF (Rich Text Format) functionality. The modular feature of Flash allows a programmer to load multiple SWF files into other SWF files. Accordingly, as discussed above, performance may be increased even further by subdividing each font into various font styles (italics, bold, italics and plain), as well as by eliminating the extended character sets.

15

20

Accordingly, to efficiently transfer fonts in the present invention, a single FLA (Flash Project File) for each font type style is created. Each of the FLA files contains a single variable print field. The font of the print field is then set to the font face and style that the style represents. Then, each FLA file is published as a font file in SWF. The SWF files are accessible by the Design Tool. All of the variable text fields in the Design Tool are loaded as part of separate Movie Clips. Each movie clip is a

25

30

named object, and the LoadMovie function is used to replace each of the Movie Clips with an instance of a font file SWF. Then, when the Design Tool receives a request to change a font of a particular field it uses the LoadMovie function to replace the current font file SWF with a new font file SWF.

5

Though the invention has been described with respect to a specific preferred embodiment, many variations and modifications will become apparent to those skilled in the art upon reading the present application. It is therefore the intention that the appended claims be interpreted as broadly as possible in view of the prior art to include all such variations and modifications.

10

WE CLAIM:

1. A method of changing an item displayed on a web page, comprising the steps of:
 - 5 loading a design tool from the web page onto a client-machine;
 - receiving a request to a change an item displayed in the design tool;
 - selectively downloading elements necessary for accomplishing the request to change an item, and
 - 10 implementing the change to the item in the client-machine to create an altered item.
2. The method of claim 1 further comprising the step of receiving an indication of user satisfaction with the appearance of the altered item.
- 15 3. The method of claim 1 further comprising the step of uploading the altered item from the client-machine to the web page.
4. The method of claim 1 further comprising the step of loading a web page containing a design tool therein from a server into a client-machine.
- 20 5. The method of claim 1 further comprising the step of uploading an item to be changed from the client-machine to the web page.
6. The method of claim 1 further comprising the step of selecting an item to be changed on the web page.
- 25 7. The method of claim 1 further comprising the step of requesting a change to a second item.

30

8. A method of transferring an item from a client-machine to a web page located on a server, the web page having a design tool, comprising the steps of:

5 receiving at a server an item uploaded from a client-machine, the item having a format;

converting the format of the item to a format compatible with the design tool to generate a design tool item; and

saving the design tool item in a memory.

10 9. The method of claim 8 further comprising the step of notifying the web page that the receiving step is complete.

10. The method of claim 8 further comprising the step of saving data regarding the item in a database.

15 11. The method of claim 8 further comprising the step of notifying a user that the step of saving is complete.

20 12. The method of claim 8 wherein the step of receiving loads the item through an Upload and Conversion module maintained on the server.

25 13. The method of claim 12 further comprising the step of moving the item from the Upload and Conversion module to the design tool.

14. A system of altering an item on a web page, comprising:
a web site;
an upload module coupled to the web site;
a design tool in communication with the web site; and
5 a memory in communication with the design tool, the web site, and
the upload system.
15. The system of claim 14 wherein the web site has an upload page.
- 10 16. The system of claim 14 wherein the web site has a design tool page.
17. The system of claim 14 wherein the upload module is an Upload
and Conversion module.
- 15 18. The system of claim 14 wherein the memory comprises a database.
19. The system of claim 14 wherein the memory comprises a file
storage.
- 20 20. The system of claim 14 wherein the design tool is executed on a
client-machine.

Figure 1 (Prior Art)

10 →

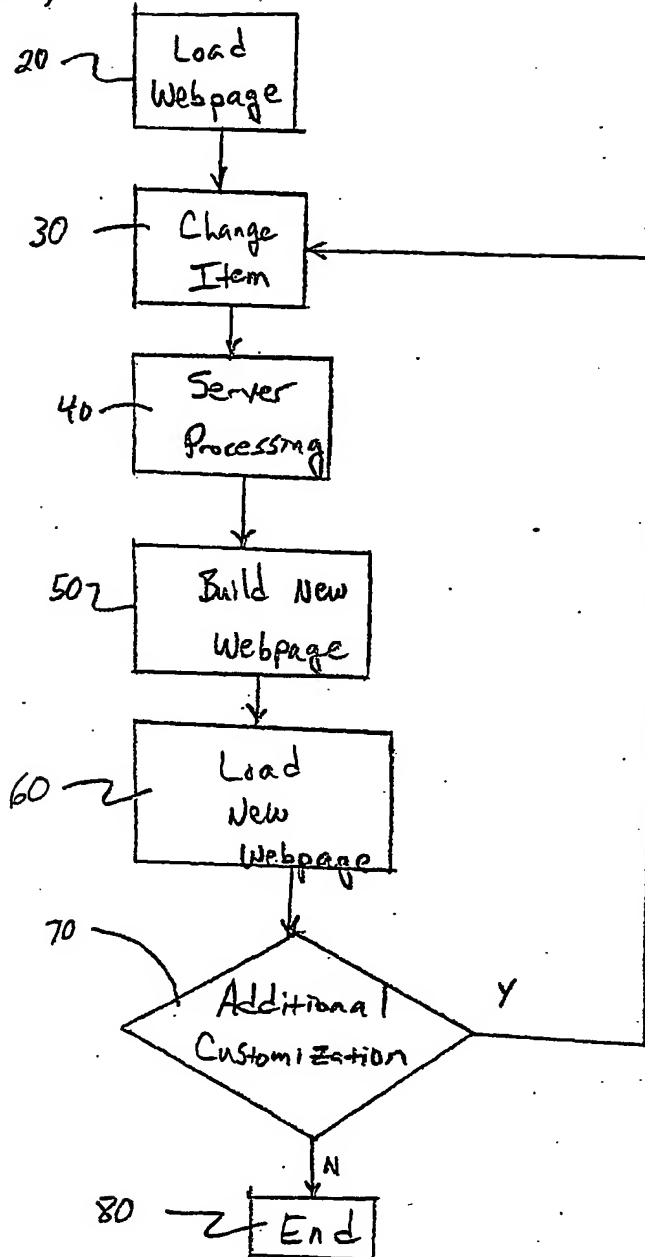


Fig. 2

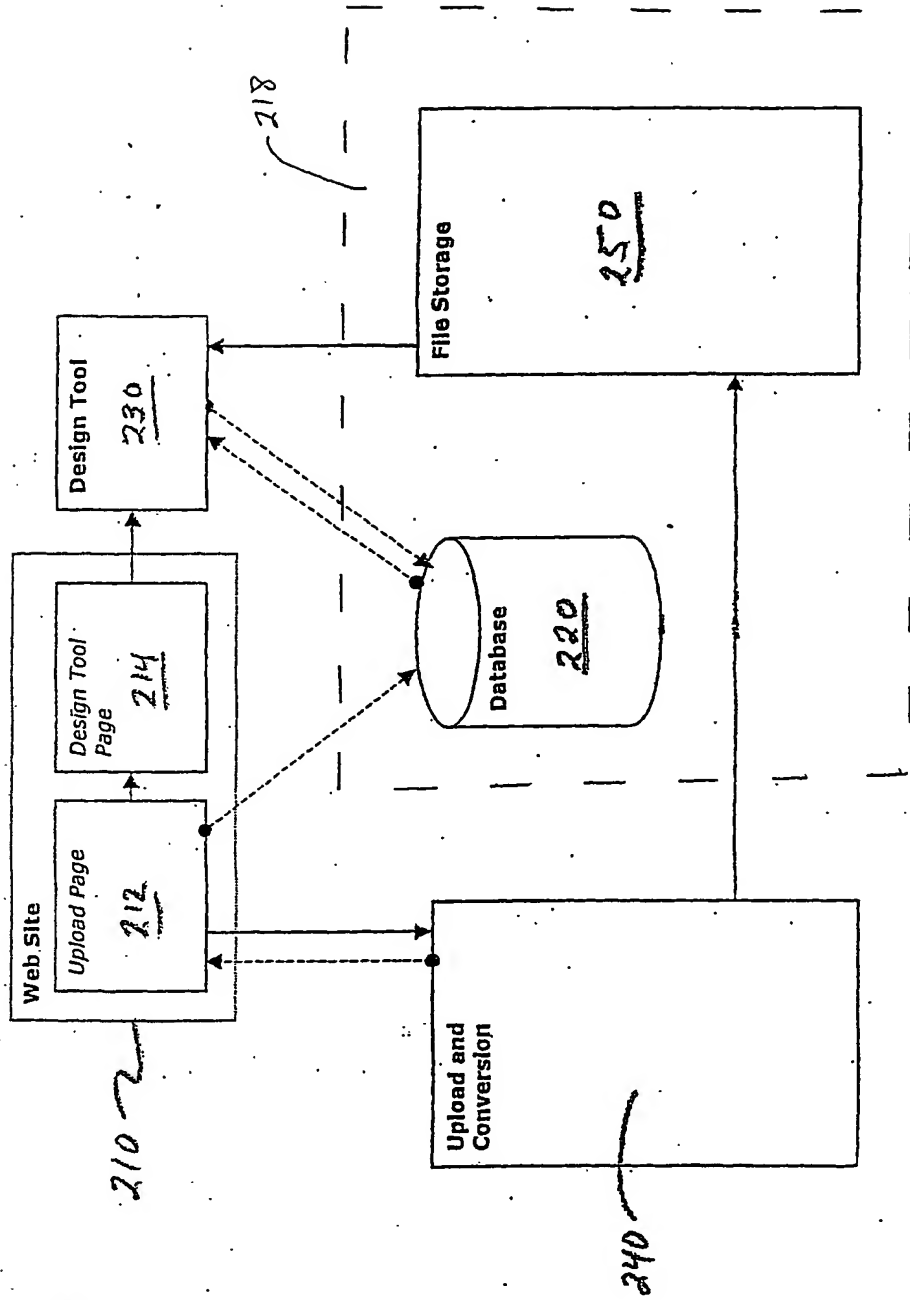


Fig. 3

300 →

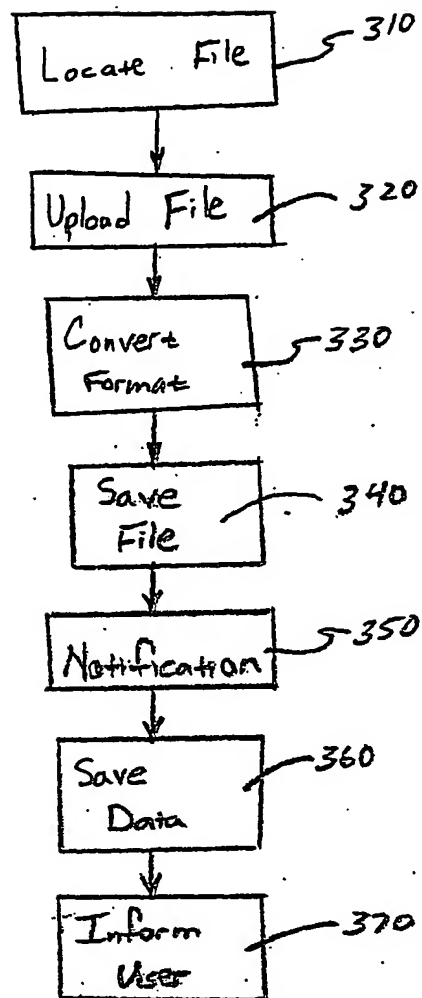
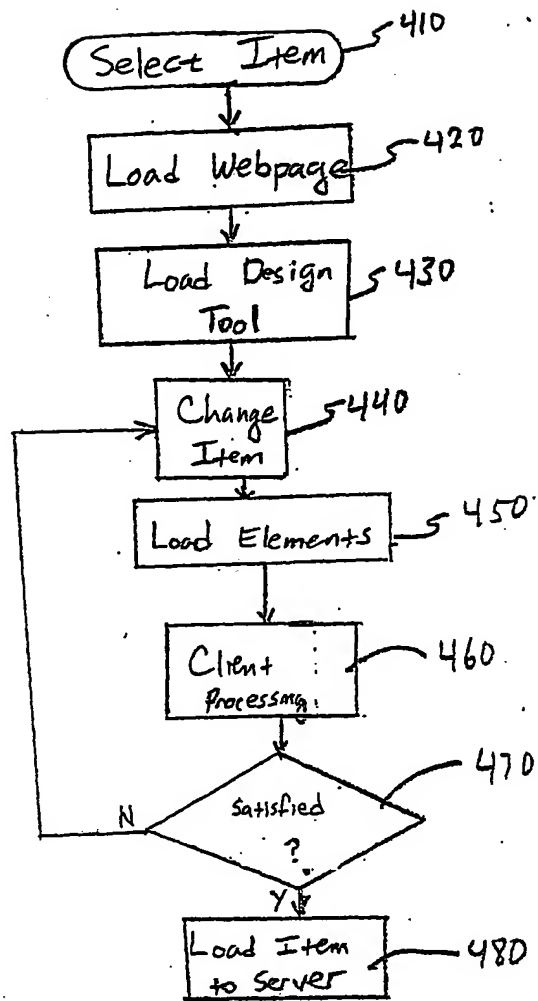


Fig. 4

400 →



INTERNATIONAL SEARCH REPORT

International Application No

PCT/US 01/09308

A. CLASSIFICATION OF SUBJECT MATTER

IPC 7 G06F17/24 G06F17/30

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC 7 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal, INSPEC, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 98 21672 A (INERGY ONLINE INC) 22 May 1998 (1998-05-22) abstract page 4, line 9 -page 6 page 10, line 6 -page 13, line 6 figures 2,3,29,49 -/-	1-7, 14-20



Further documents are listed in the continuation of box C.



Patent family members are listed in annex.

* Special categories of cited documents:

A document defining the general state of the art which is not considered to be of particular relevance

E earlier document but published on or after the international filing date

L document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

O document referring to an oral disclosure, use, exhibition or other means

P document published prior to the international filing date but later than the priority date claimed

T later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

X document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

Y document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

Z document member of the same patent family

Date of the actual completion of the international search

26 June 2001

Date of mailing of the international search report

09/07/2001

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,
Fax (+31-70) 340-3016

Authorized officer

Triest, J

INTERNATIONAL SEARCH REPORT

International Application No

PCT/US 01/09308

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
P,X	WATANABE M ET AL: "A method for supporting Web page design based on impression of Web page" PROCEEDINGS 9TH IEEE INTERNATIONAL WORKSHOP ON ROBOT AND HUMAN INTERACTIVE COMMUNICATION. IEEE RO-MAN 2000, OSAKA, JAPAN, 2, 27 - 29 September 2000, pages 13-17, XP002169587 Piscataway, NJ, USA, IEEE, USA ISBN: 0-7803-6273-X the whole document	2
X	US 5 890 170 A (SIDANA ASHMEET S) 30 March 1999 (1999-03-30) abstract column 2, line 49 -column 3, line 29	8
X	US 5 870 552 A (WILLIAMS V GEORGE W ET AL) 9 February 1999 (1999-02-09) abstract	14,18-20

INTERNATIONAL SEARCH REPORT

International Application No

PCT/US 01/09308

Patent document cited in search report		Publication date	Patent family member(s)		Publication date
WO 9821672	A	22-05-1998	AU	5356598 A	03-06-1998
			AU	9105098 A	08-03-1999
			WO	9909658 A	25-02-1999
US 5890170	A	30-03-1999	NONE		
US 5870552	A	09-02-1999	AU	705581 B	27-05-1999
			AU	5353896 A	16-10-1996
			CA	2216826 A	03-10-1996
			EP	0818013 A	14-01-1998
			JP	11503251 T	23-03-1999
			WO	9630846 A	03-10-1996

CORRECTED VERSION

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
27 September 2001 (27.09.2001)

PCT

(10) International Publication Number
WO 01/071551 A1

(51) International Patent Classification⁷: G06F 17/24,
17/30

(21) International Application Number: PCT/US01/09308

(22) International Filing Date: 22 March 2001 (22.03.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/532,713 22 March 2000 (22.03.2000) US

(71) Applicant: HCA MARKETING INC. [US/US]; 5931
Melshire Drive, Dallas, TX 75230 (US).

(72) Inventor: COLEMAN, Jason; 920 Mountain Terrace,
Hurst, TX 76053 (US).

(74) Agent: THRASHER, Steven, W.; Jackson Walker L.L.P.,
2435 N. Central Expressway, #600, Richardson, TX 75080
(US).

(81) Designated States (national): AE, AL, AM, AT, AU, AZ,
BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE,

ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP,
KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD,
MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD,
SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN,
YU, ZA, ZW.

(84) Designated States (regional): ARIPO patent (GH, GM,
KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian
patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European
patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE,
IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF,
CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:

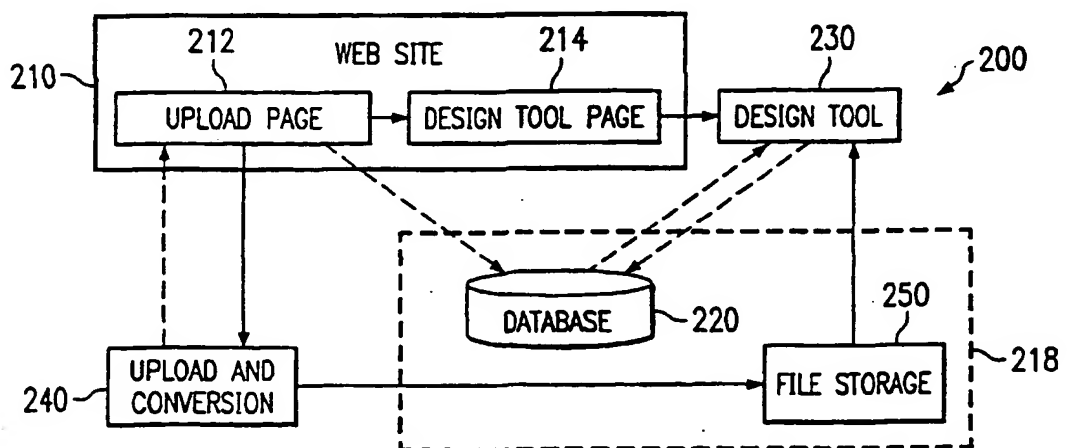
— with international search report

(48) Date of publication of this corrected version:
10 October 2002

(15) Information about Correction:
see PCT Gazette No. 41/2002 of 10 October 2002, Sec-
tion II

For two-letter codes and other abbreviations, refer to the "Guid-
ance Notes on Codes and Abbreviations" appearing at the begin-
ning of each regular issue of the PCT Gazette.

(54) Title: SYSTEM AND METHOD FOR CHANGING ITEMS ON AN INTERNET WEB SITE



(57) Abstract: The present invention provides systems and methods of changing an item on a web page. The method generally loads a design tool from the web page onto a client-machine, requests a change to an object (either uploaded to, or selected from, the web page), selectively downloads elements necessary for accomplishing the requested change to the item in the client-machine, and then performs the actual change to create an altered item. A system of altering an item on a web page includes a web site, an upload system coupled to the web site, a design tool in communication with the web site, and a memory in communication with the design tool, the web site, and the upload system.

WO 01/071551 A1

SYSTEM AND METHOD FOR CHANGING ITEMS ON AN INTERNET WEB SITE

5 FIELD OF THE INVENTION

The present invention relates generally to computer-based commerce tools, and, more particularly, to systems and methods for changing the attributes of text or graphics associated with merchandise
10 purchased over the Internet.

BACKGROUND OF THE INVENTION

As innovations in the Internet have enhanced the visual experience
15 for users of the Internet, the Internet has facilitated the sale of customized items, such as merchandise, graphics, text, or art work, for example. The sale of customized items requires vendors to supply customers with accurate, high quality images of the merchandise they may order and the customizations (or changes) the customers may apply.

20 A design tool is a software application that is used to modify the appearance of an item on a web page. In general, a design tool provides a user the ability to view an item from multiple points of view, and alter various attributes of the item, such as the items color, material, dimensions,
25 for example. Ultimately, a design tool should provide a user the visual feedback needed so that the user may determine whether or not to make a purchase. Accordingly, the design tool should present merchandise with a fidelity that is sufficient for the user to achieve meaningful understanding of what the final physical product will be like. Furthermore, it is desirous
30 that a design tool provide a user with nearly instant visual feedback of the

modifications that they are making to the merchandise, so that a user will become a customer.

5 To provide high quality images, vendors of customized items need design tools that provide visualization and customization abilities in an on-line environment. The basic method for customizing merchandise includes choosing attributes for the item, including the item's color, material, dimensions, for example, as well as the color, material, and dimensions of the item's components. Given that customized products frequently include
10 the application of customized text or customized art work, any Internet web page design tool should support the application of both text and art work. Given the need for fast response time, the dynamic application of product customization across the Internet using the prior art is particularly problematic.

15 One issue associated with product customization over the Internet is the need to reload a web page each time a single change is made to an item. Accordingly, figure 1 (Prior Art) is a flow chart of a modification algorithm 10 associated with a prior art method of changing the appearance
20 of an item on a web page. First, in a load web page step 20, a web page, typically located on a remote server, is down-loaded into a client-machine (meaning the personal computer or other computing device the user is using to alter the item). Generally, the web page will have an item which is to be changed and form element associated with it.

25 Form elements are typically icon, menu, or text-button driven commands which can change the appearance of the item. For example, one form element may allow a user to select the texture of the item (or a portion of the item). Likewise, another form element may allow the user the ability
30 to select a color for the item (or a portion of the item). Accordingly,

following the load web page step 20, the user will change the item some way in a change item step 30. In the change item step 30, the user may change the appearance of the item or alter some customizable attribute of the item, such as the items texture or material.

5

After making a change to the item, the modification algorithm 10 proceeds to a server processing step 40. In the server processing step 40, the web server uses a server - side software application to implement the desired change in order to build a new visual representation of the altered item. Following the server processing step 40, the server-side software application builds a new web page in a build new web page step 50. The new web page is built in order to display the new visual representation of the altered item. Then, in a load new web page step 60, the new web page is loaded onto the client-machine and the changed item is deployed.

15

If additional alterations are desired to change the appearance of the item, those additional alterations of the item can be achieved by following the above steps. Accordingly, in an additional customization step 70, the user may choose whether or not to make additional alterations to the item. If the user desires to make additional alterations to the item, the modification algorithm 10 returns to the change item step 30. If, however, the user does not desire to make additional alterations to the item, the modification algorithm 10 proceeds to an end step 80, and no further changes or alterations are made to the item.

25

Accordingly, each time a user wishes to make a single change to an item, the modification algorithm 10 is repeated. This means that each time the user wishes to add a new font, or change the color of even one pixel, a new image and (Hypertext Transfer Language) HTML page must be generated. This results in a new page download requirement for each

30

incremental incorporation, change, alteration or customization of the item. Thus, the user must download a number of web pages equal to the number of changes they wish to make to the item. Accordingly, the speed of the process is directly related to the number of customizations and changes in operations that user wishes to make. This is inefficient. Therefore, what is needed is a system and method of changing an item that allows for continuous, real-time feedback and realistic item representation to the user.

SUMMARY OF THE INVENTION

The present invention achieves technical advantages as a system and method of altering an item displayed on a web page. The system is
5 comprised of a software module called a Design Tool. The Design Tool is embedded in a web page which, by downloading once into a client-machine, supports the ability to make multiple changes to an item without re-loading the web page. When initially loaded into a client-machine, the Design Tool contains an image or images of an item (such as a product), as
10 well as controls for making changes to the item or items. The Design Tool provides the ability to download elements on an as needed basis in the background (preferably without the user's knowledge of the download). In addition, the system provides an upload system capable of transferring an item to be changed from the client-machine to the web page, as well as
15 memory for storing items to be altered, as well as information regarding the items. In general, the method loads the Design Tool from the web page onto a client-machine, receives a request for changing the item, loads the elements needed to alter the item, and changes the item according to the request for change. Because the changes to the item are accomplished on
20 the client-machine, visual feedback of the customized item is refreshed in real-time so that most every change is visible instantaneously to the user.

Although the new advantages of the present invention are too numerous to discuss fully here, several advantages are of particular interest.
25 First, the Design Tool executes on a client-machine, more form elements and controls may be provided to the user due to the fact that additional Design Tool functionality will not affect the speed at which a web page may be reloaded (or refreshed). Second, client-machine base applications are less likely to experience performance deterioration due to Internet
30 traffic or connection bottlenecks. Third, the present invention provides real

time, photo-realistic, images which may also be altered, rotated, or otherwise changed in real time. In addition, the overall download performance of the present invention is superior to prior art solutions by a factor equal to the number of customizations made by a user, since only a single web page must be downloaded for any number of customization steps with the present invention, while the prior art requires an additional web page to be downloaded for each alteration. Other advantages of the present invention should be apparent to those of ordinary skill in the art.

In one embodiment, the invention is a method of changing an item displayed on a web page. The method generally includes the steps of loading a design tool from the web page onto a client-machine, requesting the change of the appearance of an item displayed in the design tool, selectively downloading elements necessary for accomplishing the requested change to the item (from the web page to the client-machine), and then implementing the change to the item in real time on the client-machine to create an altered item.

In another aspect the invention includes a method of transferring to web page an item to be altered. This embodiment includes the steps of uploading an item from a client-machine to a server, converting, if necessary, the format of the item to a format compatible with a design tool, and saving the item in a file storage area.

In yet another embodiment, the present invention is a system of changing an item on a web page. The system includes a web site, an upload system coupled to the web site, a design tool in communication with the web site, and a memory in communication with the design tool, the web site, and the upload system.

BRIEF DESCRIPTION OF THE DRAWINGS

Various aspects of the invention, including alternative embodiments, can be identified by reference to the following Detailed Description of a Preferred Embodiment. Likewise, the Detailed
5 Description of a Preferred Embodiment can be better understood by reference to the drawings, in which:

Figure 1 (Prior Art) is a flow chart of a modification algorithm associated with a prior art method of changing the appearance of an item on
10 a web page;

Figure 2 illustrates one embodiment of an architecture for the design system of the present invention;

15 Figure 3 is a flow chart of an upload algorithm which may be used to upload an item to the Design Tool web page; and

Figure 4 is a flow chart of a customization algorithm used to change
20 an item in the Design Tool.

DETAILED DESCRIPTION OF A PREFERRED EMBODIMENT

The present invention provides users with real-time visual feedback of changes made to items, such as merchandise or any customizable product or service that can be visually represented or viewed over the Internet. The system provides a flexible architecture configured to enable real-time visual feedback of changes made to an item over the Internet via a web page-embedded designed tool that is downloaded to a client-machine. Likewise, the method, in one embodiment, outlines the process flow for providing real-time visual feedback of item changes.

A System Architecture

A number of components interact to provide functionality to a Design Tool, and are known collectively as a design system. In general, a design system employed by the present invention will comprise a web site, a Design Tool, an upload module, and a memory having a data base and a file storage area. Figure 2 illustrates one embodiment of an architecture for the design system 200 of the present invention.

A web site 210 functions as the basis for a user, (who could be a potential customer), interaction with the design system 200. The web site 210 includes web pages, typically written in HTML, that host a Design Tool 230, and the web site 210 has at least one page that is integrated with an upload module 240. Web site 210, includes at least a Design Tool page 214, and an upload module page 212. Preferably, the upload module 240 is built in ASP and communicates with the data base 220 through the web site 210. The user will see that the Design Tool 230 has all of those features associated with prior art design tools. Other media that can execute applications over the Internet include alternate platforms such as PDAs, Windows CE based platforms, and cellular telephones, for example.

5 The Design Tool 230 is a client-side application that is downloaded to the user web browser through the Design Tool page 214. The Design Tool is embedded in a web page, and provides the features of real-time visual feedback for changes made to items, the ability to upload items from a client-machine, the inclusion of standard items (such as art work and fonts), and allows a user to alter the appearance of the items color, alignment, font, text, size or other attribute. In addition, the Design Tool should allow for real-time dynamic sizing, positioning, and rotation of an item. Furthermore, specific construction materials may be designated for the object, and specific attributes of the material, such as its color or texture, for example, can be indicated. The present invention provides all of these functionalities and allows a user to make virtually unlimited changes to an item, while requiring the user to download only one web page.

15 The Design Tool 230 is preferably a Shock Wave Application (SWF or "Flash"). Flash is an embedded multimedia format that can be played directly in a web page browser. Flash is preferred because it delivers high fidelity content at a low-bandwidth cost and can deliver an application in a small file size. Furthermore, ActionScript, the native scripting language of Flash, provides a language in which a design tool may be written. Accordingly, ActionScript enables the Design Tool 230 to be configured to dynamically manipulate text, objects, and other attributes of items. Of course, other types of script may be used and other applications may be employed to build Design Tool 230.

25 Another advantage provided by building the Design Tool 230 with Flash is that Flash can be authored modularly so that multiple SWF files can be embedded in other SWF files, thus allowing the embedding of additional fonts and objects into the Design Tool. In addition, Flash may

be streamed, meaning that a SWF file may be displayed in a web page browser even though parts of the SWF file are not yet completely downloaded. This allows additional design tool file components to be downloaded without a user being aware that the download is taking place.

5 In addition, streaming allows the Design Tool 230 to add additional files to the client-machine without loading additional web pages.

The Design Tool 230 can be created in any self-contained, low bandwidth media that is capable of interactivity and scripting features (similar to Flash). Two such platforms capable of providing similar features are JAVA and Visual Basic (VB). However, applications created in JAVA or VB or similar platforms that perform the same or similar tasks as the Design Tool typically have large file sizes. Thus, they are typically prohibitively large for acceptable download performance. However, as

10 Internet access bandwidth increases over time, it is conceivable that in the future Internet connection speeds may be capable of supporting design tool applications developed in JAVA, VB, or other similar platforms available for the design tool process. In addition, it is conceivable that software manufacturers and developers will distribute new applications (similar to

15 Flash) capable of implementing the Design Tool 230.

The upload module 240, implemented specifically in the present invention as an Upload and Conversion module is typically a server-side software program. The upload module 240 comprises applications that

25 allow a user to have the capacity to transfer items, such as image files, stored in the local client-machine memory, to memory 218 at the server (such as file storage 250). The upload module 240 may convert an uploaded item to a format compatible with the Design Tool 230 so that the items may be changed in the Design Tool according to user preferences.

30

More specifically, the upload module 240 is part of the vendor web site, and allows a user to browse their local computer client-machine to select an item for uploading to the web page. The web page 210 then stores the item in a section of memory 218 known as file storage 250 and
5 associates the item with the user who uploaded it in a database 220. Preferably, the item is stored in a format which can be embedded in the Design Tool 230.

Accordingly, the upload module 240 includes a conversion
10 application to convert files from known formats, such as JPG, TIF, for example into a format which is embeddable into the Design Tool 230, such as Flash. The Upload and Conversion module uses Macromedia Generator to embed graphic images into a flash format, while the uploading and conversion of an item is achieved in three distinct parts: an upload and
15 storage stage, a conversion to PNG format stage, and an embedding into SWF formats.

In the upload and storage stage, Web Super Goo's Image Goo is employed as the server- side component of the upload and storage process.
20 The upload and storage process is driven by the web page's ASP (Active Server Page) script. When Image Goo completes the upload and storage process, it then starts the conversion process. The conversion process generates a Portable Network Graphic (PNG) version of the file. The PNG format is needed because it is capable of supporting a transparency attribute
25 when it is imported into Flash.

Accordingly, Image Goo opens a file to be converted, sets a background color to be the "transparent" color, and saves the file in a PNG format. When the upload and storage, and conversion processes are
30 complete, the ASP page is notified by Web Super Goo. Then, the ASP

page will call Macromedia Generator to begin the embedding process. ASP code in the web page instructs Macromedia Generator to use specific templates and to embed specific files.

5 Macromedia Generator then creates a new SWF file and embeds the
uploaded item into the SWF file. The Macromedia Generator saves the
SWF file in memory 218 in file storage 250 and notifies the ASP page that
the item has been stored in file storage 250. The ASP page then records the
path name and ownership information regarding the uploading item in the
10 data base 220 contained in memory 218. The ASP page also notifies the
user that the process of uploading and storage is complete and may allow
the user the opportunity to upload another item or to navigate to other parts
of the web page. Accordingly, the SWF files created by Macromedia
Generator are ready to be inserted into the Design Tool 230.

15 Of course, the Macromedia Generator may in future versions have
its functionality extended to achieve all of the above processes. It is also
possible that other software manufacturer may distribute applications that
are capable of performing some or all of these processes as well. In
20 addition, Web Super Goo and Macromedia Generator are capable of being
controlled by means other than an ASP page. For example, they could be
driven by PERL or other CGI language.

25 Memory 218 provides the database 220 and the file storage 250.
The database 220 is capable of storing information, and is used to store
information about the uploaded items, customer accounts, customized
items, such as merchandise, created by the Design Tool 230. The file
storage 250 is used to store files that may be needed by the Design Tool
230. In addition, the file storage 250 holds the font files and image files as

well as the art uploaded by a user so that these items may be pasted to the upload module 240.

5 It is desirable that as items are stored, that they can be changed into a standardized file format which can be used by the vendors who actually create merchandise. Of course, this data could also be transferred to a proprietary format if the vendor so desires. Accordingly, items should be capable of being stored in formats traditional in the digital printing industries, bordering industries, screening, engraving, embossing and debossing, dye cutting, and other manufacturing industries. Furthermore, it is desirable that files, such as font files, be capable of downloading to a client-machine on an as needed basis and of downloading in the background while the Design Tool 230 is running on the client-machine. Accordingly, font files and image files should be stored as modules.

15 In addition, art files may have colors which are associated, or indexed, to be transparent—this means that they display on top of the product image without displaying the rectangular background shapes that accompany any type of raster (BMP type) image. Also, specific type styles of a particular font will have their own module meaning that a font's italic, bold, underline, and bold-italic styles will each have their own module.

20 The LoadMovie function of Shockwave Flash allows a SWF file to be dynamically loaded into other SWF files. Accordingly, user may control the type and specific source of image files they would like to embed through applications developed in Action Script.

25 The file storage 250 stores media files in the original uploaded format as well as other formats used throughout the conversion process. The file storage area 250 does not have to be a single device or depository

for files, and, in fact, may function more efficiently if it is made up of several unique depositories. For example, a separate file server may be used to implement a portion of file storage 250 used to store uploaded image items. Then, as the size of memory used by certain types of items grows, additional servers may be added to implement the file storage 250.

Database 220 is preferably implemented as a Microsoft SQL server database. To convert items into a format useable by a vendor, several applications must be employed. Accordingly, Macromedia Generator is used to dynamically build graphic files based on input data (information about the item). However, Macromedia Generator creates content for the Internet, it is not enabled to create the formats desired by vendors, such as encapsulated postscript EPS, PDF (Portable Document Format), AI (Adobe Illustrator), or PJTF (Portable Job Ticket Format), for example.

An offline system including Macromedia Generator and other software is used to convert the data recorded by a customer into a format usable by a printer or manufacturer using templates appropriate for the item type. Accordingly, the SWF file that is produced by the Macromedia Generator is saved in file storage 250. Next, an automated version of the Flash Editor is employed to convert SWF format files into EPS. The Flash Editor creates a new FLA file, and imports the SWF file then exports the EPS file. The EPS file is specified in this example because it is a widely accepted format used by graphic professionals, and is readily adaptable to printers. However, it should be understood, that a SWF formatted file may be transferrable into other formats.

A Method of Implementation

The present invention allows the user to customize an item by changing (altering the appearance of, or adding art or textual objects to) an

item selected from a catalog on the Design Tool web page, or which is uploaded from a client-machine. Figure 3 is a flow chart of an upload algorithm 300 which may be used to upload an item to the Design Tool web page running on a server. In general, the upload algorithm 300 uploads an item as a computer file, converts the file into a format embeddable into the Design Tool, and then saves the file into the design system memory.

More specifically, the upload algorithm 300 begins with a locate file step 310 in which the user locates on the client-machine a file the user would like to use as an art object or item in the Design Tool. Next, in an upload file step 320, the file is specifically transferred from the client-machine to the server running the design system, and the file is sent to the upload module. The uploaded file step 320 may load a file directly from the client-machine to an upload module 240, such as the Upload and Conversion module.

Next, the upload module converts the format of the uploaded file in a convert format step 330. In the convert format step 330, the uploaded file is altered from its original format into a format which is embeddable by the Design Tool 230. Once the uploaded file is converted into a format usable by the Design Tools, the file compatible with the Design Tool is stored in file storage in memory in a save file step 340. Next, in a notification step 350 the upload module is notified that the transfer of the item is complete. Then, following the notification step 350, data about the item is saved in a database in a save data step 360. After data about the item is saved in a database, the user is informed that the transfer of the item from the client-machine to the web page is complete in an inform user step 370.

A method of changing an item through a Design Tool located and running on a client-machine may be better understood by expressing the

method as an algorithm. Figure 4 is a flow chart of a customization algorithm 400 used to change an item in the Design Tool. First, in a select item step 410, a user will either choose to change an item he has uploaded or may choose to change an item which already exists on the web page.

5 Once the item to be changed is selected in select item step 410, the customization algorithm 400 proceeds to a load web page step 420. In the load web page step 420, the user accesses the Design Tool page 214 which automatically loads the Design Tool 230 onto the client-machine the user is using then a load Design Tool 430. Next, in a change item step 440, the

10 user may use the Design Tool controls to make changes to the item.

Next, in a load element step 450 the Design Tool cooperates with the web page to selectively download elements (such as modules) necessary for accomplishing the requested change to the item. Accordingly, these

15 elements are transferred from the web page to the client-machine. Then, in a client processing step 460, the Design Tool 230 running on the client-machine implements the changes to the item requested by the user to create an altered item. Thus, the changes are immediately made to the item, and the user of the Design Tool immediately sees the changes on the client-

20 machine without a new web page downloading.

If the user desires to make an additional change to the item, all the user has to do is make the change and the customization algorithm 400 will automatically process the change request by automatically returning the

25 change item step 440, and the additional changes will be processed, and indicated by the no, or "N" decision in a satisfied step 470. If the user is satisfied with the item at this point, thus desiring no further changes, the user may indicate so by entering in an indication of satisfaction by actuating an icon, selecting a menu command, or some other means. Alternatively,

30 user satisfaction with the item may be assumed if the user does not interact

with the item for a predetermined period of time, such as three minutes. This decision by the user to not make additional changes to the item is represented as a "Y" in the satisfied step 470. Following the satisfied step 470, if the user is satisfied with the item, the customization algorithm 400 proceeds to a load item to server step 480.

5

In the load item to server step 480 the altered item with all of its changes is loaded into the web page in a single step. Then, the web page may be refreshed in the browser on the client-machine. Accordingly, the present invention displays and makes all user changes to an item, and displays those changes to the user while requiring the loading of only one web page (whereas, in the prior art, a new web page would have to be loaded following each incremental change made to the item).

10

15

To quickly and efficiently transfer fonts between the web page and a client-machine, the modular features of Flash are adapted to work with the font face control of the variable field to emulate RTF (Rich Text Format) functionality. The modular feature of Flash allows a programmer to load multiple SWF files into other SWF files. Accordingly, as discussed above, performance may be increased even further by subdividing each font into various font styles (italics, bold, italics and plain), as well as by eliminating the extended character sets.

20

25

Accordingly, to efficiently transfer fonts in the present invention, a single FLA (Flash Project File) for each font type style is created. Each of the FLA files contains a single variable print field. The font of the print field is then set to the font face and style that the style represents. Then, each FLA file is published as a font file in SWF. The SWF files are accessible by the Design Tool. All of the variable text fields in the Design Tool are loaded as part of separate Movie Clips. Each movie clip is a

30

named object, and the LoadMovie function is used to replace each of the Movie Clips with an instance of a font file SWF. Then, when the Design Tool receives a request to change a font of a particular field it uses the LoadMovie function to replace the current font file SWF with a new font file SWF.

5

Though the invention has been described with respect to a specific preferred embodiment, many variations and modifications will become apparent to those skilled in the art upon reading the present application. It is therefore the intention that the appended claims be interpreted as broadly as possible in view of the prior art to include all such variations and modifications.

10

WE CLAIM:

1. A method of changing an item displayed on a web page, comprising the steps of:
 - 5 loading a design tool from the web page onto a client-machine;
 - receiving a request to a change an item displayed in the design tool;
 - selectively downloading elements necessary for accomplishing the request to change an item, and
 - 10 implementing the change to the item in the client-machine to create an altered item.
2. The method of claim 1 further comprising the step of receiving an indication of user satisfaction with the appearance of the altered item.
- 15 3. The method of claim 1 further comprising the step of uploading the altered item from the client-machine to the web page.
4. The method of claim 1 further comprising the step of loading a web page containing a design tool therein from a server into a client-machine.
- 20 5. The method of claim 1 further comprising the step of uploading an item to be changed from the client-machine to the web page.
6. The method of claim 1 further comprising the step of selecting an item to be changed on the web page.
- 25 7. The method of claim 1 further comprising the step of requesting a change to a second item.

30

8. A method of transferring an item from a client-machine to a web page located on a server, the web page having a design tool, comprising the steps of:

5 receiving at a server an item uploaded from a client-machine, the item having a format;

converting the format of the item to a format compatible with the design tool to generate a design tool item; and

saving the design tool item in a memory.

10 9. The method of claim 8 further comprising the step of notifying the web page that the receiving step is complete.

10. The method of claim 8 further comprising the step of saving data regarding the item in a database.

15

11. The method of claim 8 further comprising the step of notifying a user that the step of saving is complete.

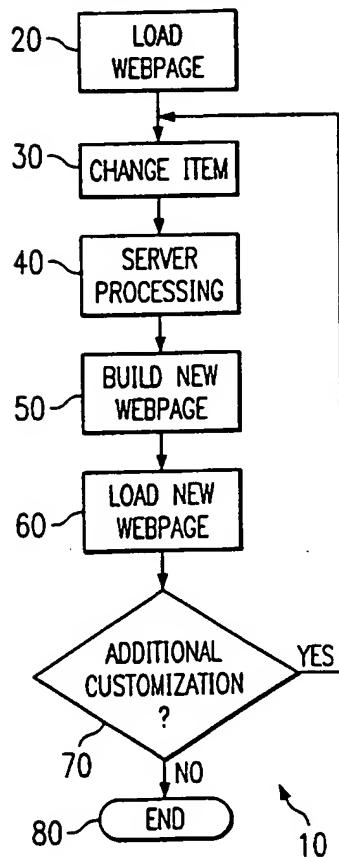
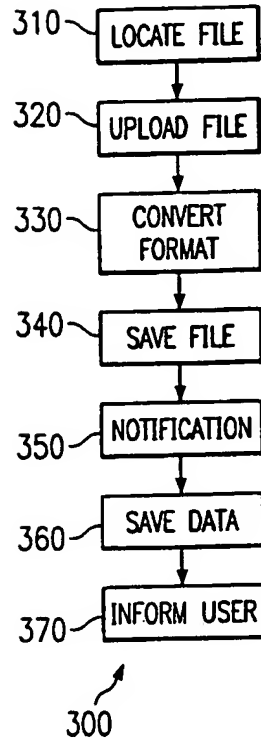
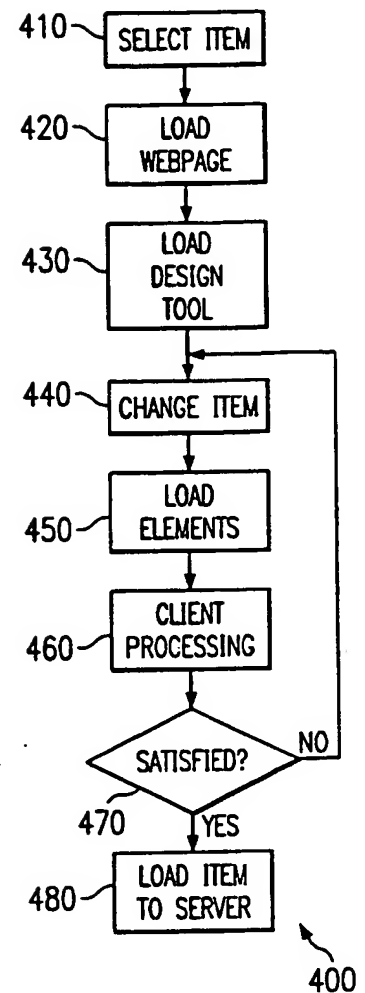
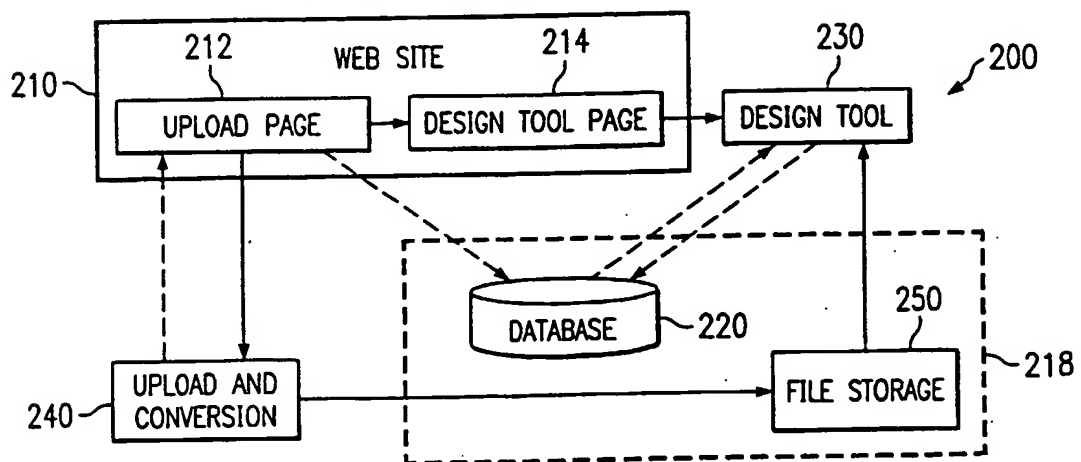
20 12. The method of claim 8 wherein the step of receiving loads the item through an Upload and Conversion module maintained on the server.

13. The method of claim 12 further comprising the step of moving the item from the Upload and Conversion module to the design tool.

25

14. A system of altering an item on a web page, comprising:
a web site;
an upload module coupled to the web site;
a design tool in communication with the web site; and
5 a memory in communication with the design tool, the web site, and
the upload system.
15. The system of claim 14 wherein the web site has an upload page.
- 10 16. The system of claim 14 wherein the web site has a design tool page.
17. The system of claim 14 wherein the upload module is an Upload
and Conversion module.
- 15 18. The system of claim 14 wherein the memory comprises a database.
19. The system of claim 14 wherein the memory comprises a file
storage.
- 20 20. The system of claim 14 wherein the design tool is executed on a
client-machine.

1/1

FIG. 1
(PRIOR ART)*FIG. 3**FIG. 4**FIG. 2* ✓

INTERNATIONAL SEARCH REPORT

International Application No

PC 17 US 01/09308

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G06F17/24 G06F17/30

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC 7 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal, INSPEC, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 98 21672 A (INERGY ONLINE INC) 22 May 1998 (1998-05-22) abstract page 4, line 9 -page 6 page 10, line 6 -page 13, line 6 figures 2,3,29,49 — -/-	1-7, 14-20

☒ Further documents are listed in the continuation of box C.☒ Patent family members are listed in annex.

* Special categories of cited documents:

A document defining the general state of the art which is not considered to be of particular relevance

E earlier document but published on or after the international filing date

L document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

O document referring to an oral disclosure, use, exhibition or other means

P document published prior to the international filing date but later than the priority date claimed

T later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

X document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

Y document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

8 document member of the same patent family

Date of the actual completion of the international search

26 June 2001

Date of mailing of the international search report

09/07/2001

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl.
Fax: (+31-70) 340-3016

Authorized officer

Triest, J

INTERNATIONAL SEARCH REPORT

International Application No
PL., JS 01/09308

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT		
Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
P, X	<p>WATANABE M ET AL: "A method for supporting Web page design based on impression of Web page"</p> <p>PROCEEDINGS 9TH IEEE INTERNATIONAL WORKSHOP ON ROBOT AND HUMAN INTERACTIVE COMMUNICATION. IEEE RO-MAN 2000, OSAKA, JAPAN, 2,</p> <p>27 - 29 September 2000, pages 13-17, XP002169587</p> <p>Piscataway, NJ, USA, IEEE, USA</p> <p>ISBN: 0-7803-6273-X</p> <p>the whole document</p> <p style="text-align: center;">---</p>	2
X	<p>US 5 890 170 A (SIDANA ASHMEET S)</p> <p>30 March 1999 (1999-03-30)</p> <p>abstract</p> <p>column 2, line 49 -column 3, line 29</p> <p style="text-align: center;">---</p>	8
X	<p>US 5 870 552 A (WILLIAMS V GEORGE W ET AL) 9 February 1999 (1999-02-09)</p> <p>abstract</p> <p style="text-align: center;">-----</p>	14, 18-20

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No

PCT/US 01/09308

Patent document cited in search report		Publication date	Patent family member(s)	Publication date
WO 9821672	A	22-05-1998	AU 5356598 A	03-06-1998
			AU 9105098 A	08-03-1999
			WO 9909658 A	25-02-1999
US 5890170	A	30-03-1999	NONE	
US 5870552	A	09-02-1999	AU 705581 B	27-05-1999
			AU 5353896 A	16-10-1996
			CA 2216826 A	03-10-1996
			EP 0818013 A	14-01-1998
			JP 11503251 T	23-03-1999
			WO 9630846 A	03-10-1996

Form PCT/ISA/210 (patent family annex) (July 1992)